Indian Institute of Technology Patna
CS1101- Foundations of Programming

Lab8: : High-Level to Low-Level Programming: Concepts and Applications

Date: 6-10-2025

Download the files and work in a separate directory named **Lab8**. For this lab we will be using online simulator

https://cpulator.01xz.net/?sys=nios-de1soc

Task1: Familiarize yourself the with following functions (clear_screen(), write_pixel(), write_char () and test sample program given.

```
/* set a single pixel on the screen at x,y
 * x in [0,319], y in [0,239], and colour in [0,65535]  */
void write_pixel(int x, int y, short colour) {
  volatile short *vga_addr=(volatile short*)(0x08000000 + (y<<10) + (x<<1));
  *vga_addr=colour;
}

/* use write_pixel to set entire screen to black (does not clear the character buffer) */
void clear_screen() {
  int x, y;
  for (x = 0; x < 320; x++) {
    for (y = 0; y < 240; y++) {
      write_pixel(x,y,0x0000);
    }
  }
}
/* write a single character to the character buffer at x,y
 * x in [0,79], y in [0,59]
 */
void write_char(int x, int y, char c) {
  // VGA character buffer
  volatile char * character_buffer = (char *) (0x09000000 + (y<<7) + x);
  *character_buffer = c;
}

int main () {

   clear_screen();
   return 0;
}
```

Exercise1 : Write function to set color of the screen and test.

```c
/* VGA colors */
#define WHITE 0xFFFF
#define YELLOW 0xFFE0
#define RED 0xF800
#define GREEN 0x07E0
#define BLUE 0x001F
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define GREY 0xC618
#define PINK 0xFC18
#define ORANGE 0xFC00

int main () {

   clear_screen();
   setColor(WHITE);
   setColor(YELLOW);
   setColor(RED);

   return 0;
}
```

(20 Points)

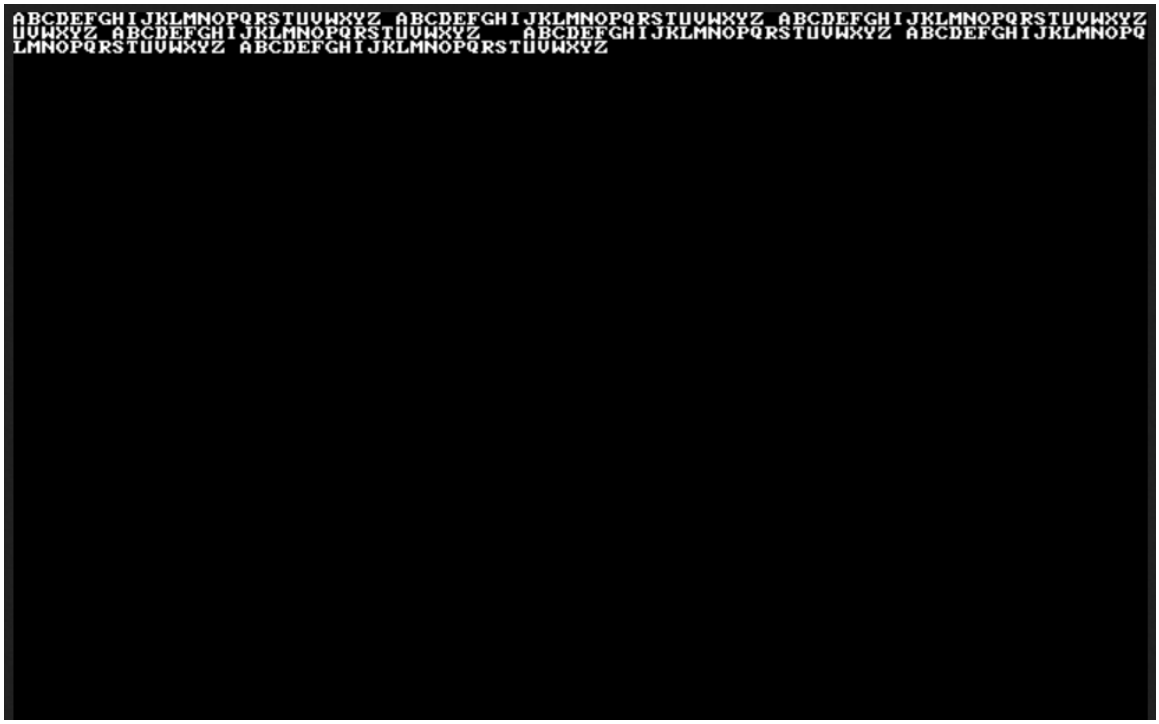
Task2: Using the following print hello world to the screen.

```
void write_pixel(int x, int y, short colour) {
  volatile short *vga_addr=(volatile short*)(0x08000000 + (y<<10) + (x<<1));
  *vga_addr=colour;}
/* use write_pixel to set entire screen to black (does not clear the character buffer) */
void clear_screen() {
  int x, y;
  for (x = 0; x < 320; x++) {
    for (y = 0; y < 240; y++) {
      write_pixel(x,y,0);
    }  }}

void write_char(int x, int y, char c) {
  // VGA character buffer
  volatile char * character_buffer = (char *) (0x09000000 + (y<<7) + x);
  *character_buffer = c;
}

int main () {
   clear_screen();
   int x;
 // Write Hello, world!
   char* hw = "Hello, world! 1";
   x = 15;
   while (*hw) {
     write_char(x, 10, *hw);
     x++;
     hw++;
   }
   return 0;
}
```

Exercise2 : Using the above display “ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ…..” to the VGA screen.



(20 Points)

Task3: Draw a straight line in red across the screen centre and Draw a "diagonal" line in green (20 Points)

```
int main () {

   clear_screen();
   int x;

   for (x=0;x<320;x++)
   {
      // Draw a straight line in red across the screen centre
      write_pixel(x, 59, 0xf800);
      // Draw a "diagonal" line in green
      if (x<240)
         write_pixel(x, x, 0x07e0);
   }
```

Exercise3 : Write program to test multiple lines and test.

Task 4: Study the following functions to draw a circle and line.
(20 Points)

```
#include<math.h>
void write_pixel(int x,int y,short color)
{
        if(x<320 && y<240)
        {
        volatile short *vga_addr = (volatile short*)(0x08000000+(y<<10)+(x<<1));
        *vga_addr = color;
        }
}

//Write a pixel
void write_char(int x,int y,char c)
{
        if(x<79&&y<59)
        {
        volatile char* char_buff = (char*) (0x09000000+(y<<7)+x);
        *char_buff = c;
        }
}


//Clear screen
void clear_screen(short color)
{
        int x,y;
        for(x=0;x<320;x++)
        {
                for(y=0;y<240;y++)
                {
                        write_pixel(x,y,color);
                }
        }
}

//Clear character buffer
void clear_text()
{
        int x,y;
        for(x=0;x<79;x++)
```

```
	{
		for(y=0;y<59;y++)
		{
			write_char(x,y,' ');
		}
	}
}
//Complete clear function
void clear(short color)
{
	clear_screen(color);
	clear_text();
}

void draw_line(int x1,int y1,int x2,int y2,short color)
{
	//Slope of the line
	if(x2!=x1)
	{
		float slope = (float)(y2-y1)/(float)(x2-x1);
		//increment (is the line going from right to left or vice versa)
		int inc;
		//(if x1>x2)
		if(x1>x2)
		{
			inc =-1;
		}
		else
		{
			inc =+1;
		}


		for(int x = x1;x!=x2;x+=inc)
		{
			int y = y1 + (int)round(slope*(float)(x-x1));
			write_pixel(x,y,color);
		}
	}
	else
	{

		int inc;
		if(y1>y2)
		{
			inc =-1;
```

```
                }
                else
                {
                        inc =+1;
                }

                for(int y= y1;y!=y2;y+=inc)
                {

                        write_pixel(x1,y,color);
                }
        }
}

void draw_rect(int x1,int y1,int width,int height,short color)
{
        for(int i=y1;i<y1+height;i++)
        {
                draw_line(x1,i,x1+width,i,color);
        }
}

void draw_circle(int cx,int cy,int r,short color)
{
        int ymin = cy - r;
        int ymax = cy +r;

        for(int y=ymin;y<=ymax;y++)
        {
                int dy = abs(y-cy);
                int dx = sqrt(r*r - dy*dy);

                draw_line(cx-dx,y,cx+dx,y,color);
        }
}


int main()
{
        clear(0);
        draw_circle(140, 140, 100,0x001F);
        return 0;


}
```

Exercise4 : Write C program to draw multiple circle and different locations

Task 5: Write C program to display your own pattern and text.

In Record

Task 5

Task 1, 2, 3,4 , and 5 : Demonstrate your work to TAs.